Parameter Optimization for Evolutionary Algorithm – Quadratic Assignment Problem

S. Muenvanichakul[1], C. Chanpilom[2]

[1]Department of Industrial Engineering, Kasetsart University Sriracha Campus, Chonburi 20230, E-mail: sirirat.m@ku.ac.th
[2] Department of Industrial Engineering, Kasetsart University Sriracha Campus, Chonburi 20230, E-mail: chakapan@eng.src.ku.ac.th

***Abstract* – Quadratic Assignment Problem (QAP) is a NP-hard problem. In order to eliminate binary decision variables and assignment constraint, the problem is formulated in a constraint logic programming in which the binary decision variables are now part of the constraint. Evolutionary algorithm is used to find the solution of the constraint logic programming. Mutation rate and population size are the parameters in the algorithm. Response Surface Technique is used to optimize those parameters. Response considered in this study is assignment cost (QAP objective value). Result shows that QAP in constraint logic programming substantially reduces computational intensity allowing the modest-size problems can be solved in Microsoft Excel 2010. Parameter optimization by means of response surface approach to locate optimal mutation rate and population size shows a promising trend. However, further study should be carried out to establish a solid ground for the parameter optimization methodology proposed in this study.**

Keywords – Quadratic Assignment Problem (QAP), Constraint Logic Programming, Evolutionary Algorithm, Response Surface Technique and Parameter Optimization.

1. Introduction

Quadratic Assignment Problem (QAP) is a decision problem with a goal to minimize an assignment cost of *N* facilities over *N* locations subjected to the assignment constraints. Traditionally, QAP is formulated as a Mixed-Integer Linear Programming (MILP) [1] which is NP-hard problem and has remained one of the greatest challenges in combinatorial optimization e.g. [2]. The problem can alternatively be casted as a logical inference problem allowing an implementation of constraint logic programming (CLP), which potentially offers several advantages [3]. In the framework of CLP, the assignment problem is in the form of a permutation problem subjected to *all-different* global constraint. The size of this permutation problem is substantially reduced in comparison to original MILP problem leading to the size of the problem that is more manageable. It may be possible to obtain QAP solution by means of existing solution methods to QAP in CLP framework.

An evolutionary algorithm with *all-different* constraint available in Microsoft Excel 2010 solver is adopted in this study to solve QAP in CLP framework. The objective of this study is to optimize two tuning parameters in the evolutionary algorithm, namely population size and mutation rate, by means of response surface analysis via central composite experiment. Many recent studies [4-7] support our approach – parameter tuning in algorithms can be achieved by means of design of experiment and response surface analysis.

The article starts by stating QAP in CLP framework in section 2. Section 3 briefly discusses evolutionary algorithm in Microsoft Excel 2010 Solver. Response surface analysis and the responses used in this work are considered in section 4. Result and discussion are given in section 5. We conclude and discuss future work in section 6.

2. Quadratic Assignment Problem in Constraint Logic Programing Framework

A mathematic model of quadratic assignment problem (QAP) in constraint logic programming (CLP) framework is given in [3] as followed**:**

Minimization

$$\text{Assignment Cost} = \sum_{n}^{i=1}\sum_{n}^{k=1} F_{ik}\, D_{y_i y_k} \qquad (1)$$

Subject to

$$\text{all} - \text{different}\ \{y_1, y_2, \dots, y_n\} \qquad (2)$$

where

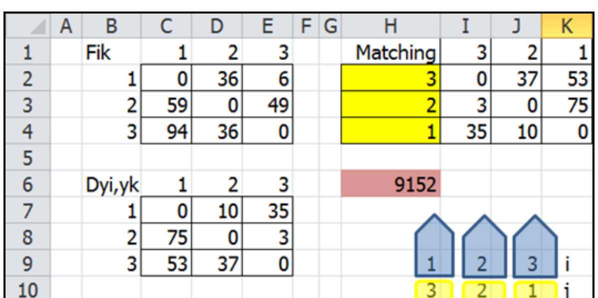
- *n* represents the number of facilities/locations (1,2,..,*N*)
- y_n represents the facility that is located at y_n location; must be a permutation of indices
- $C_{i,y_i,k,y_k} = F_{i,k} \times D_{y_i,y_k}$ represents the assignment cost of assigning facility *i* to location y_i and facility *k* to location y_k
- $F_{i,k}$ represents the flow between facility *i* to facility *k*
- D_{y_i,y_k} represents the cost between location y_i to location y_k .

There are many software/systems that support the CLP such as Prolog, Xpress, ECLiPSe etc. However, they require programming skill to code the problems; it may not be accessible to untrained users. This work prefers software/program that requires minimum or no programming skills ;thereby adopting evolutionary engine in Microsoft Excel 2010 Premium Solver Platform.

Since the maximum allowable number of decision variables allowed in Premium Solver Platform is 8000 [8], the Solver can handle QAPs only up to the size of $N = 9$, if the problem is casted as linearized QAPs in MILP problems. This is because the linearization in MILP framework introduces a new decision variable Y_{ijkl} that has the size of N^4 in addition to an original decision variable X_{ij} that has the size of N^2 [2,9]. If, however, the problems are casted in CLP formulation, there is no need to introduce the new decision variable Y_{ijkl}. In this case, the modest-size problems can be reasonably accommodated by Premium Solver. This kind of problems is generally considered as a computationally non-trival task, if the problems are casted as MILP problems [9].

3. Evolutionary Algorithm in Microsoft Excel

Evolutionary algorithm is a subset of a generic population-based metaheuristic optimization algorithm. The concept of evolutionary algorithms is inspired and derived from evolutionary processes found in nature such as reproduction, mutation, recombination and selection. In Premium Solver, the decision variable i.e. the assignment is encoded in a series of bit strings and successively updated as it goes though successive generations (or iterations) in order to achieve the optimal assignment yielding the minimum assignment cost defined in Equation (1) and subjected to *all-different* constraint in Equation (2).



	A	B	C	D	E	F	G	H	I	J	K
1		Fik	1	2	3			Matching	3	2	1
2		1	0	36	6			3	0	37	53
3		2	59	0	49			2	3	0	75
4		3	94	36	0			1	35	10	0
5											
6		Dyi,yk	1	2	3			9152			
7		1	0	10	35						
8		2	75	0	3						
9		3	53	37	0						
10											

Cell	Formula	Copied to
I1	=H2	-
J1	=H3	-
K1	=H4	-
I2	=INDEX($C$7:$E$9,$H2,I$1	I2:K4
H6	=SUMPRODUCT(C2:E4,I2:K4)	-

Figure 1 Portion of spreadsheet illustrating the implementation of QAP with $N = 3$. Formula in different cells also given.

Fig. 1 illustrates the portion of the spreadsheet where QAP with $N = 3$ is implemented. Formula related to the calculation of the objective function defined in Equation (1) is also given in the figure. Fig. 2 shows parameters set in *Solver Parameters Window* in the spreadsheet for $N = 3$ problem. To solve larger-size problems, we modify cell indices in this spreadsheet and in *Solver Parameters Window*.

Solver Parameters:
Set Objective: **$H$6**
To: **Min**
By Changing Variable Cells: **$H$2:$H$4**
Subject to the Constraints: **$H$2:$H$4 = AllDifferent**
Select a Solving Method: **Evolutionary**
Options: Evolutionary:
Convergence: **0.0001**
Mutation Rate: "be studied"
Population Size: "be studied"
Random Seed: **0**
Maximum Time without Improvement: **30**

Figure 2 Solver parameters setting[1]

Evolutionary algorithm requires the inputs of two parameters namely population size and mutation rate. Note that for high-levels of mutation rates, the algorithm is closer to a random process. Finding optimal values for population size and mutation rate is challenging and will be further studied below.

4. Response Surface Analysis

Population size and mutation rate for optimal condition are determined by means of response surface model. Central composite experimental design [10] is used to specific the values of population size and mutation rate to be used in the response surface model. The response considered is the assignment cost defined in Equation (1). QAP studied in [11] is used as a benchmark for our parameter design approach. In order to come up with descend levels of population size and mutation rate to start, we perform line searches of these two parameters for a given problem size *N*. We start by fixing the population size to be 200 and perform a line search to find two levels of mutation rates that produce the two highest probabilities to find the best solutions as shown in Table 1. To find two best population size levels, we start from the mutation rate that produces the highest probability to find the best solution determined in the previous step. Then we perform another line search to determine two population size levels that produce the two highest probabilities to find best solution shown in Table 1. Note that the population size of 200 that we use to start to process is arbitrary. However, the size is chosen such that it is the largest population size that we can effort to run a line search of mutation rates in a reasonable of time.

Our experiments are designed in Minitab 16. Table2 shows the values of population size and mutation rate derived from central composite design for problems with different *N*. Note that additional center points are added to ensure the rotatability of the experimental design [10]. In

[1] The unit of "Maximum Time without Improvement" is seconds.

total, there are 13 plans for population size and mutation rate given in Table 2 for each problem size, *N*. For each plan, the Premium Solver in Microsoft Excel 2010 performs the operation for 30 iterations

Table 1 The two levels of population sizes and mutation rates for problems with different *N*.

Case	N	Parameter Levels of			
		Mutation Rate		Population Size	
		Low	High	Low	High
1	12	0.4	0.8	200	500
2	14	0.8	0.9	500	700
3	15	0.1	0.5	200	600
4	16b	0.6	0.9	250	450
5	17	0.35	0.8	500	700
6	18	0.2	0.7	450	900
7	21	0.15	0.7	300	700

Table 2 The plans with different population size and mutation rate from central composite design for problems with different *N*.

Plan no.	Master		N=12		N=14		N=15		N=16b		N=17		N=18		N=21	
	Parameter		Parameter		Parameter		Parameter		Parameter		Parameter		Parameter		Parameter	
	(1)*	(2)*	(1)*	(2)*	(1)*	(2)*	(1)*	(2)*	(1)*	(2)*	(1)*	(2)*	(1)*	(2)*	(1)*	(2)*
1	-1	-1	0.4	200	0.8	500	0.1	200	0.6	250	0.35	500	0.2	450	0.15	300
2	-1	1	0.4	500	0.8	700	0.1	600	0.6	450	0.35	700	0.2	900	0.15	700
3	1	-1	0.8	200	0.9	500	0.5	200	0.9	250	0.8	500	0.7	450	0.7	300
4	1	1	0.8	500	0.9	700	0.5	600	0.9	450	0.8	700	0.7	900	0.7	700
5	0	-1.41	0.6	139	0.85	459	0.3	117	0.75	209	0.575	459	0.45	357	0.425	217
6	0	1.414	0.6	562	0.85	741	0.3	683	0.75	491	0.575	741	0.45	993	0.425	783
7	-1.41	0	0.317	350	0.779	600	0.017	400	0.538	350	0.257	600	0.096	675	0.036	500
8	1.414	0	0.883	350	0.921	600	0.583	400	0.962	350	0.893	600	0.804	675	0.814	500
9	0	0	0.6	350	0.85	600	0.3	400	0.75	350	0.575	600	0.45	675	0.425	500
10	0	0	0.6	350	0.85	600	0.3	400	0.75	350	0.575	600	0.45	675	0.425	500
11	0	0	0.6	350	0.85	600	0.3	400	0.75	350	0.575	600	0.45	675	0.425	500
12	0	0	0.6	350	0.85	600	0.3	400	0.75	350	0.575	600	0.45	675	0.425	500
13	0	0	0.6	350	0.85	600	0.3	400	0.75	350	0.575	600	0.45	675	0.15	300

* (1) = Mutation Rate, (2) = Population Size

5. RESULT

The problems with different sizes, *N*, studied in [11] are used as a benchmark for our parameter optimization procedure proposed here. As a preliminary study, this study is restricted to the problem sizes that are less than 21. From available dataset in QAPLIB, the sizes of the problems considered include *N = {12, 14, 15, 16b, 17, 18, 21}*. Note that there are two cases for *N = 16*. Problem instances and solutions are given in Quadratic Assignment Problem Library (QAP Library) [9]. Table 3 lists optimal assignment cost and layout, y_n, for problems with different sizes, *N*.

Problem instances in [11] obtained from [9] are input into Microsoft Excel 2010. For each problem size, *N*, 13 plans with different population size and mutation rate given in Table 2 are carried out in the spreadsheet. For each plan, the solver performs the operation for 30 iterations. Other parameters in the evolutionary algorithm are set according to those shown in Fig. 2.

For the first task, we explore the ability of evolutionary algorithm in Microsoft Excel 2010 over population size and mutation rate space to solve QAP. So for each problem size, *N*, we search for the best minimum assignment costs for all 13 plans and iterations. We measure and compare frequencies and probabilities that the evolutionary algorithm finds the optimal solutions. The result is show in Table 4.

Table 3 Optimal assignment cost and layout, y_n, for problems with different *N* [9,11].

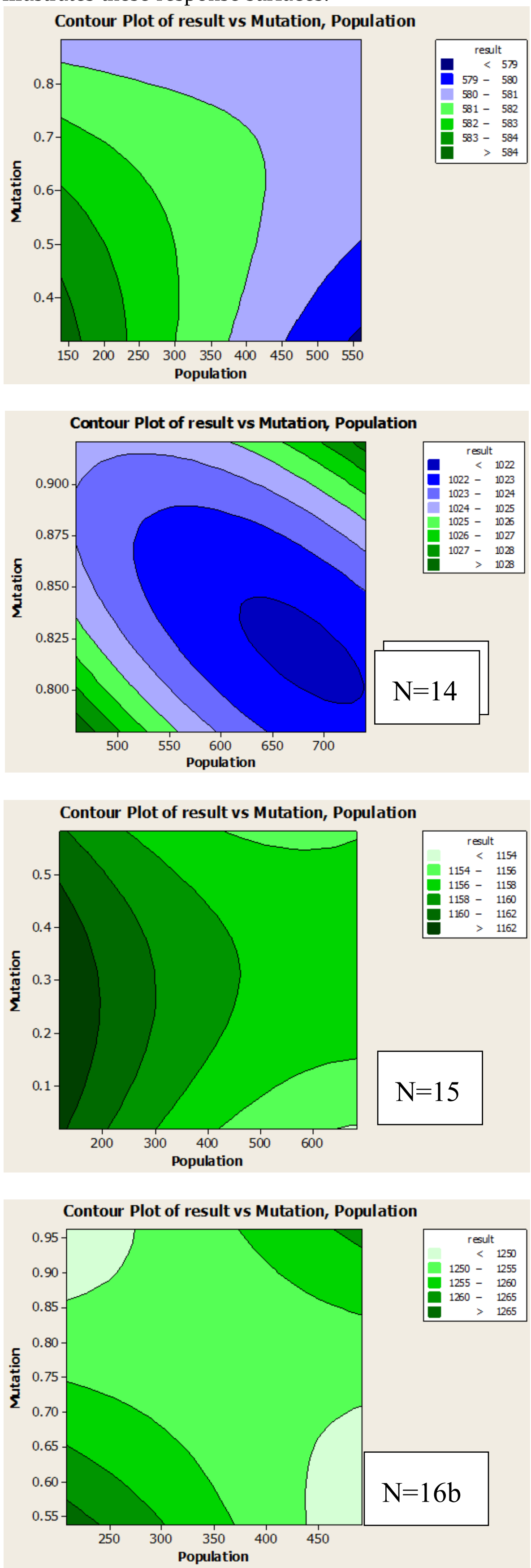
N	Assignment Cost	Layout
12	578	(12,7,9,3,4,8,11,1,5,6,10,2)
14	1014	(9,8,13,2,1,11,7,14,3,4,12,5,6,10)
15	1150	(1,2,13,8,9,4,3,14,7,11,10,7,3,14,6,1,5)
16b	1240	(16,12,13,8,4,2,9,11,15,10,7,3,14,6,1,5)
17	1732	(16,15,2,14,9,11,8,12,10,3,4,1,7,6,13,17,5)
18	1930	(10,3,14,2,18,6,7,12,15,4,5,1,11,8,17,13,9,16)
21	2438	(4,21,3,9,13,2,5,14,18,11,16,10,6,15,20,19,8,7,1,12,17)

Table 4 Characteristics of solutions with different problem sizes, *N*, from Microsoft Excel 2010.

N	Total Assignment Cost (unit cost)			Optimal Solution found		Average Executing Time (min.)
	Max.	Min.	Avg.	Freq.	pdf.	
12	600	**578**	581.37	217	0.56	0:01:35
14	1050	**1014**	1023.33	45	0.12	0:01:46
15	1200	**1150**	1158.57	23	0.06	0:01:33
16b	1294	**1240**	1253.49	189	0.48	0:01:16
17	1794	**1732**	1750.18	42	0.11	0:02:01
18	2010	**1930**	1962.62	23	0.06	0:01:41
21	2568	**2438**	2479.09	2	0.01	0:02:47

As the problem sizes, *N*, increased, the frequencies and probabilities that evolutionary algorithm finds optimal solution should be less and eventually become zero. From the result, the probabilities of finding optimal solution are around 0.1 except for *N* = 12, 16 and 21. For *N* = 12 and 16, the probabilities are about in 0.5, while for *N* = 21, the probability is only 0.01. It is clear that the evolutionary algorithm can find optimal solution of QAP even the size of the problem is up to 20. Note that Microsoft Excel can handle the problem size that is less than 9 i.e. *N* = 9 due to the limit of maximum allowable number of decision variables, if the problems are casted in MILP formulation as discussed in section 2. Study [12] also points out this fact. As a result, it is not possible to compare executing times between MILP and CLP formlations for modest to large scale problems. Clearly, QAP in CLP framework shows a great promise as an alternative to QAP in MILP framework. Average executing times for different problem sizes, *N*, are also given in Table 4. With this alternative formulation, Microsoft Excel 2010 takes only about a few minutes to find optimal solutions, which is reasonable.

Response surfaces for different *N* are constructed in Minitab 16 based on 13 plans with different population size and mutation rate with 30 iterations for each plan. Fig. 3 illustrates these response surfaces.



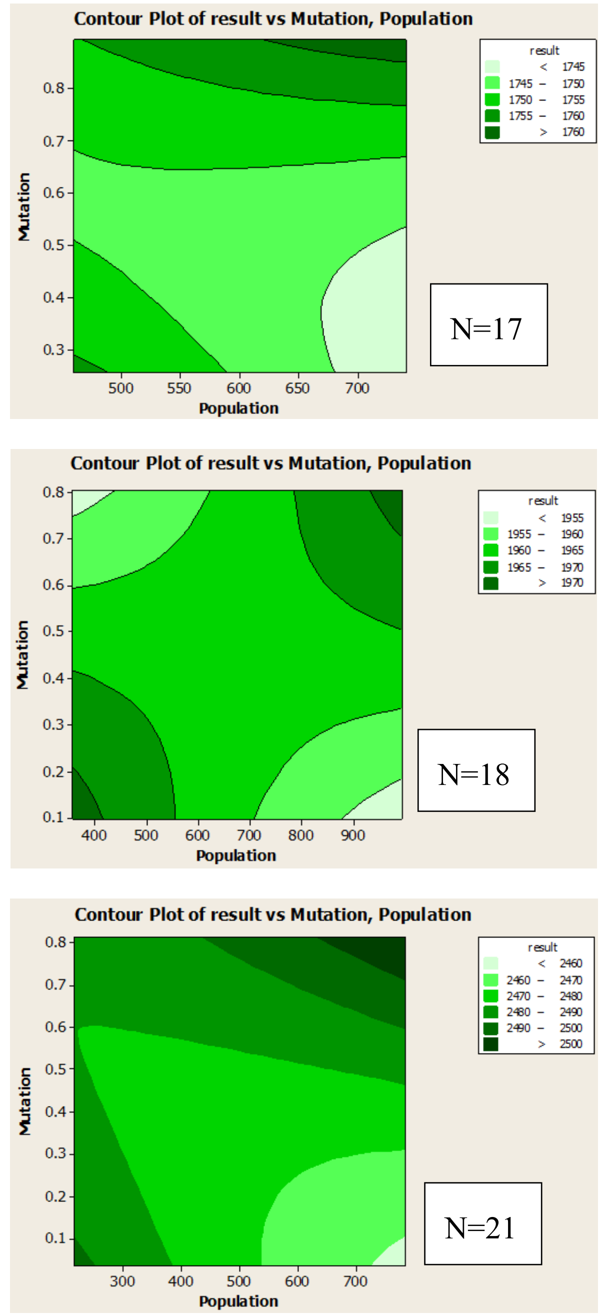


Figure 3 Response surfaces for different *N* constructed in Minitab 16 based on 13 plans with different population size and mutation rate with 30 iterations for each plan

Note that resulting response surface models are second-order. The optimal population sizes and mutation rates are determined by response optimizer available in Minitab 16. Table 5 shows optimal population sizes and mutation rates together with optimal solutions obtained from response surfaces. Bear in mind that the optimal solutions obtained from response surface models are not the same as the true optimal solutions shown in Table3 because response surfaces based on regression though data points and it is impossible to arrive at the true optimal solution. We also present the desirability function obtained from Minitab 16 in Table 3 to underscore the quality of our estimated optimal solution in comparison to the true optimal solution. At this point, we only concern about the optimality of the solution and there is other output to be compromised with the optimality of the solution. We assume that the optimal population size and mutation rate will produce lowest value of response (assignment cost), and produce high probability that the evolutionary algorithm will find true optimal assignment cost. This assumption has not been systematically verified in this study and will be subjected to future work. At this point, we discuss the result based on this assumption.

Table 5 Optimal population sizes, mutation rates, optimal solutions and desirability functions obtained from response surfaces for different problem size, *N*.

N	Mutation Rate	Population Size	D Desirability	Y Assignment Cost (unit cost)	QAPLIB Optimal Assignment Cost (unit cost)
12	0.317	562	0.972	578.787	578
14	0.819	678	0.843	1021.841	1014
15	0.017	683	0.932	1153.854	1150
16b	0.538	491	0.898	1246.349	1240
17	0.285	741	0.891	1741.348	1732
18	0.096	993	0.771	1951.941	1930
21	0.036	783	0.840	2457.396	2438

For given parameter ranges, only response surface for N = 14 gives a clear (local) minimum response. For other N, minimum responses occur at the edge of domain suggesting that we might have to shift the parameter range to new local minimum zone that will produce a lower minimum response (and higher desirability function).

6. Conclusion and Future Work

We explore the possibility of using evolutionary algorithm in Microsoft Excel 2010 to solve quadratic assignment problem (QAP) in constraint-logic programming (CLP) framework. With all-different constraint from assignment constraint, QAP in CLP framework becomes a permutation problem. The problem has become more manageable allowing optimization process on Microsoft Excel 2010 possible. The challenge is now on how to find two optimal parameters (population size and mutation rate) for evolutionary algorithm. The main focus for this work is to perform parameter optimization for these two parameters for evolutionary algorithm using response surface approach. Central composite design in Minitab 16 is used to find the test matrix. Parameter optimization is carried out in Minitab 16 using Response optimizer.

Result shows that CLP framework greatly simplifies QAP and allows evolutionary algorithm in Microsoft Excel 2010 to solve the problem up to modest size i.e. $N \leq 21$ within a reasonable executing time. This is much more promising than a solution method based on QAP in MILP framework. Response surfaces for different problems are constructed and used in parameter optimization. The trend of the result obtained from the method shows a promising sign but future work in the following areas is recommended to systematically verify the parameter optimization method and generalize it to general QAP:

1. Since the optimal assignment costs showed are from response surface models, it is not clear what the statistical characteristics (such as average, standard deviation or probability density function) of best assignment costs obtained from evolutionary algorithm at given optimal parameters will be. Comparison against other parameter plans obtained from central composite design will also bring us more confidence on the level of optimality achieved by the method.
2. From our result, it seems that the range of parameters chosen in this study is not the best possible one. We can either try to a method to find a better range of parameters or apply a combination of steepest-descent method with successive response surfaces to move parameter range toward optimal parameter values, see for example in [13]
3. Once a solid grounding for parameter optimization has be established, wider classes of QAP available in QAP Library [9] should be tested in order to fine tune and generalize the method to general QAP.

7. Acknowledgment

SM gratefully acknowledges Assoc. Prof. Peerayuth Charnsethikul for his initial suggestion to use new feature “all-different constraint” in Microsoft Excel 2010 to solve QAP in CLP and many insightful discussion. Much of the calculation was done by Mr. Bawonwong Kitcharoen.

8. References

[1] E.L. Lawler, “The quadratic assignment problem,” *Manage. Sci. 9*, pp. 586-599. DOI: 10.1287/mnsc.9.4.586, 1963.

[2] S. Muenvanichakul, and P. Charnsethikul, “Benders’ decomposition based heuristics for large-scale dynamic quadratic assignment problems,” *J. Comput. Sci.*, *5*, pp. 64-70. DOI: 10.3844/jcssp.2009, 2009.

[3] J. Hooker, *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. John Wiley & Sons, 1st Ed., 2000.

[4] S.F. Attar, M. Mohammadi, R. Tavakkoli-Moghaddam, and S. Yaghoubi, “Solving a new multi-objective flexible flowshop problem with limited waiting times and machine-sequence-dependent set-up time constratints,” *Int. J. Comput. Integ. M.*, 27, pp. 450-469, 2014.

[5] M.B. Abello and Z. Michalewicz, “Implicit memory-based technique in solving dynamic scheduling problems through response surface methodology – Part I: Model and method” *Int. J. Intell. Comput. Cybern.*, 7(2), pp. 114-142, 2104.

[6] O.A. Adbul-Rahman, M. Munetomo and K. Akama, “An adaptive parameter binary-real coded genetic algorithm for constraint optimization problems: Performance analysis and estimation of optimal control parameters,” *Inform. Sciences*, 233, pp. 54-86, 2013.

[7] M. Zandich, E. Mozaffari, and M. Gholami, “A Robust Genetic Algorithm for Scheduling Realistic Hybrid Flexible Flow Line Problems”, *J. Intell Manuf.*, 21, pp. 731-743, 2010. DOI: 10.1007/s10845-009-0250-5.

[8] FrontlineSolvers 2014, *Standard Excel Solver – Dealing with Problem Size Limits*. Available from: <

http://www.solver.com/standard-excel-solver-dealing-problem-size-limits> [24 November 2014].

[9] R.E. Burkard, E. Çela, S.E. Karisch, and F. Rendl, *QAPLIB – A quadratic assignment problem library*, 2011. Retrieved from http://anjos.mgi.polymtl.ca/qaplib/.

[10] D.C. Montgomery, *Design and analysis of Experiments*. Wiley, 8th Ed., 2012.

[11] T.E. Vollman, C.E Nugent, and R.L. Zartler, "A computerized model for office layout," *J. Ind. Eng.*, *19*, pp. 321-327, 1968.

[12] C. Chanpilom, "Development of solution method for quadratic assignment problem," Undergraduate Industrial Engineering Project Report, Faculty of Engineering at Si Racha, Kasetsart University Siracha Campus (in Thai), 2013.

[13] R.H. Myers, D.C. Montgomery, and C.M. Anderson-Cook, *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. Wiley, 3rd Ed., 2009.

9. BIOGRAPHIES

Dr. Sirirat Muenvanichakul is a lecturer in Faculty Engineering at Si Racha, Kasetsart University Siracha Campus. Her research focuses on operations research, optimization algorithms including heuristic search, evolutionary/genetic algorithms and tabu search. She holds a B.Eng., M.Eng. and D.Eng. in Industrial Engineering from Kasetsart University.

Mr. Chakapan Chanpilom has recently joined Faculty Engineering at Si Racha, Kasetsart University Siracha Campus as a lecturer. His research focuses on operations research and optimization. He received his B.Eng. (Hons) from Kasetsart University